Use of HSM with Relational Databases

Randall Breeden, John Burgess and Dan Higdon
FileTek, Inc.
9400 Key West Avenue
Rockville, MD 20850
jgb@filetek.com
Tel: 301-251-0600
Fax: 301-251-1990

Hierarchical Storage Management (HSM) systems have evolved to become a critical component of large information storage operations. They are built on the concept of using a hierarchy of storage technologies to provide a balance in performance and cost. In general, they migrate data from expensive high-performance storage to inexpensive low-performance storage based on frequency of use. The predominant usage characteristic is that frequency of use is reduced with age and in most cases quite rapidly. The result is that HSM provides an economical means for managing and storing massive volumes of data.

Inherent in HSM systems is system managed storage, which has the system performing most of the work with minimum operations personnel involvement. This automation is generally extended to include:

- Backup and recovery
- Data duplexing to provide high availability
- Catastrophic recovery through use of off-site storage

Types of HSM

HSM can be broken into two main categories based upon the level of the objects that are accessible through the HSM system: file level and record level.

File Level

Most of today's HSM systems operate on a magnetic disk file level basis. In these HSM systems, when data is migrated off magnetic disk, the associated directory entry remains while the actual data is moved down the hierarchy. When the end-user or end-user application needs the migrated data, the file containing the data is opened, and the data is migrated back to magnetic disk. For example, if transaction information for a deposit that occurred a year ago is required, the HSM system copies the entire file back to the magnetic layer, and then the application extracts the specific information it needs.

Record Level

The second type of HSM system operates on a record level access basis. In these HSM systems, data is written with one or more keys or a record number. Then when the end-user or end-user application needs information, the file containing the required data is opened, a key or record number is supplied, and the associated record is transferred. The main difference between file and record level HSM systems is that in record level HSM systems, data can be accessed directly from the storage media without having to be restored to the magnetic layer first. This is particularly useful when storing billions of small objects such as user transactions, phone calls, and statements.

The following table compares the performance of file and record level access HSM systems.

Action	**File HSM (seconds)**	**Record HSM (seconds)**
Mount Tape	15	15
Copy Data to Mag (500 Mbytes)	100	NA
Perform high-speed search for block	NA	10
Select 1 Record	1	1
Total	116	26

The preceding table shows a significant performance advantage for record level HSM when only a small object is needed. This is even more significant when optical disks are used instead of tape. This performance improvement can make the difference between being able to provide an online response versus a batch and call back response. Another significant advantage is that the storage drives used to support the accesses are in use much less for each request enabling many more requests to be processed per day.

Record level HSM has been used in mainline storage management for a number of years for microfiche replacement, online report viewing, IBM VSAM archiving and application-based database extension.

HSM In Databases

HSM has seen little use with databases. Only small databases are built on the file system enabling the use of file level HSM. In these cases, the delay required to return the file (table) usually makes it impractical.

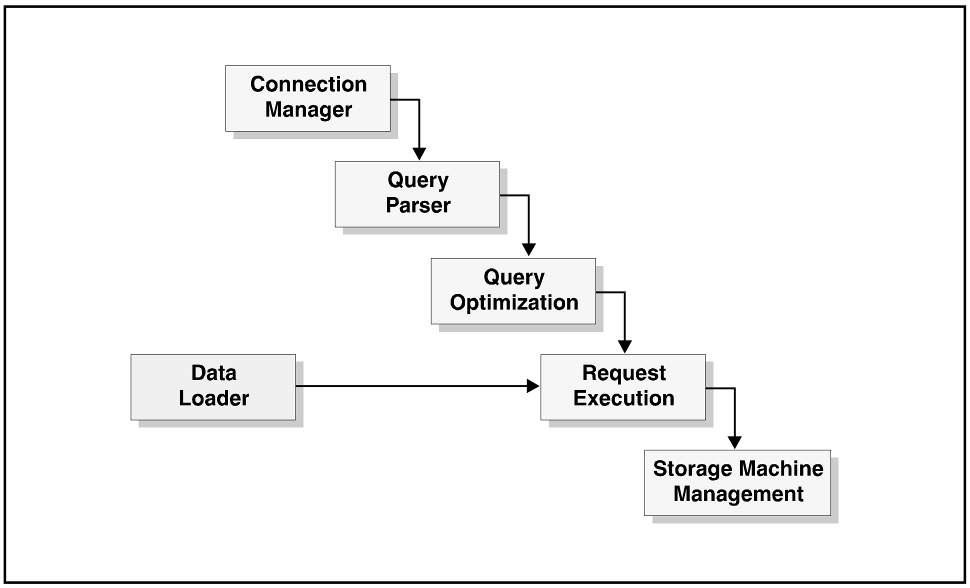
StorHouse™ System

StorHouse is the first relational database system that was developed to be fully integrated with a record level HSM system (DB/HSM™). It is built on the proven base of the FileTek® Storage Machine® (SM) system, which has been in operational use in nearly one hundred sites for managing close to 200 TBytes of online storage.

StorHouse has a high volume data loader for Direct Channel loads from mainframes and FTP loads from the network. It can process 10s of GBytes of table data per day and concurrently build all required indexes. It can build massive tables spanning many years and holding 10s of TBytes. Both hash and value indexes are supported to enable fast exact match retrievals and range-based retrievals. Indexes are multilevel and can reside separately in the storage hierarchy. This enables indexes to reside on high performance storage (RAID or optical) while data resides on less expensive storage (optical or tape).

StorHouse contains its own SQL query processor, optimizer, execution manager and database gateways. The SQL query processor, optimizer and the Storage Machine ensure that SQL queries are processed such that minimal use of robotics (optical and tape) is required. This support includes the use of large magnetic disk performance buffers that enable the storage of 100s of GBytes of the most active portions of indexes or tables to further enhance performance. These performance sensitive capabilities are extremely important because database queries executed against very large databases (VLDBs) can be very demanding.

The following diagram illustrates the various StorHouse components.



Database gateways provide access to StorHouse from many different database systems. Today, StorHouse supports DB/2® using DRDA™, EDA/SQL™ and ODBC™. In the future, StorHouse will support several other yet-to-be-announced middleware standards. StorHouse and the gateways provide for full sharing of data from different database environments. For example, data stored from MVS® DB/2 can be accessed by ORACLE® environments. This open query capability enables ad hoc queries to be processed online in support of all operational systems.

StorHouse will have a high volume data extractor that can access 100s of GBytes per day for bulk loading into RDBMSs or analytical tools. This will provide data for decision support applications whether they be OLAP or Data Mining.

Summary

StorHouse provides a low-cost storage alternative for RDBMS data using the Storage Machine's automatic managed storage hierarchy. StorHouse eliminates the need for separately archiving SQL databases to tape and supports SQL access to very large and ultra large databases. With standard protocol access from a variety of computing platforms, StorHouse expands the media options by migrating RDBMS data tables from expensive mainframe DASD and client/server magnetic disk to lower-cost reusable or permanent storage. By providing concurrent access to relational data from multiple host environments, applications can truly share data without having to maintain multiple copies. This improves service, reduces the cost of magnetic storage, frees up existing magnetic storage for other applications and eliminates the use of tape as an additional archive method for database data. Furthermore, network and channel activity is reduced because StorHouse returns only the requested result set.

The following diagram shows StorHouse's role in an information technology environment.